An Interop Publication

ConneXions®

The Interoperability Report

September 1994 Volume 8, No. 9

ConneXions —
The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

ConneXions is published monthly by Interop Company, a division of ZD Expos, 303 Vintage Park Drive, Foster City, California, 94404–1138, USA.
Phone: +1 (415) 578-6900
Fax: +1 (415) 525-0194
E-mail: connexions@interop.com

Subscription hotline: 1-800-575-5717
or +1-502-493-3217

ISSN 0894–5926

From the Editor

Welcome to Atlanta and *NetWorld+Interop 94,* the fourth stop on our 1994 World Tour. This issue of *ConneXions* is being provided to all conference, tutorial and exhibition attendees while supplies last. We are pleased to offer a special 20% discount on all new subscriptions. To take advantage of this offer, simply complete the enclosed card and drop it in the mail. For those of you unfamiliar with this publication, *ConneXions* is a monthly technical journal covering all aspects of computer networking and interoperability. It is the companion journal to the conference, and has been published by Interop Company since the first Interop conference in 1987. For an index of back issues, send e-mail to: `connexions@interop.com.`

Our first article this month is a look at cryptographic algorithms for providing security in computer networks. William Stallings describes conventional encryption and provides an overview of the most important algorithms in this first installment. Next month, public-key cryptography and secure hash functions will be covered.

Security is of great importance to the rapidly growing Internet and a number of engineering efforts related to security are underway within the *Internet Engineering Task Force* (IETF). Jim Galvin of Trusted Information Systems gives an overview of these activities. The IETF is the primary standards development body for the Internet, and if you are interested in participating in its work you should make note of the information at the end of Dr. Galvin's article.

The *Windows Sockets* Application Program Interface (API) 1.1 is a sockets-style transport interface for the Microsoft Windows family of operating systems. Inspired by the need for a binary-compatible interface to TCP/IP stacks under Windows, a great deal of momentum has built around this API. J. Allard of Microsoft describes work currently being done to extend Windows Sockets to version 2.0.

George Abe of Cisco Systems describes the characteristics of what is known as "Corporate IP" as distinct from commercial IP service. Both commercial and corporate IP will continue to play an important role in the global information infrastructure.

The *Desktop Management Task Force* (DMTF) is a consortium of vendors who are working to define operating system independent, open desktop management interfaces. The architecture is described in an article by John McConnell.

You will find more information on all of these topics in our extensive conference and tutorial program, and we hope you will continue to receive updates on emerging technology through *ConneXions—The Interoperability Report.* Enjoy your week in Atlanta!

Back to Basics:

Cryptographic Algorithms

Part I: Conventional Cryptography

by William Stallings

Introduction

A growing proportion of the applications and protocols used over the Internet either have significant security-related features or have as their primary purpose the provision of some security facility. At the application level, examples include e-mail security (Privacy Enhanced Mail, PEM; Pretty Good Privacy, PGP), network management (Simple Network Management Protocol version 2, SNMPv2), and remote authentication (*Kerberos*). A common feature of all of these applic - ations and protocols is the use of cryptographic algorithms to implement particular security services. The many such algorithms in use fall into three categories: conventional encryption algorithms, public-key cryptography algorithms, and secure hash functions. This month's article deals with conventional encryption and provides an overview of important algorithms in each category. Next month, public-key cryptography and secure hash functions are covered.

Conventional encryption

Conventional encryption, also referred to as *symmetric encryption* or *single-key encryption,* was the only type of encryption in use prior to the development of public-key encryption. It remains by far the most widely used of the two types of encryption.

In conventional encryption, the original intelligible message, referred to as *plaintext,* is converted into apparently random nonsense, referred to as *ciphertext.* The encryption process consists of an *algorithm* and a *key.* The key is a value independent of the plaintext that controls the algorithm. The algorithm will produce a different output depending on the specific key being used at the time. Changing the key changes the output of the algorithm. Once the ciphertext is produced, it is transmitted. Upon reception, the ciphertext can be transformed back to the original plaintext by using a decryption algorithm and the same key that was used for encryption.

The security of conventional encryption depends on several factors. First, the encryption algorithm must be powerful enough so that it is impractical to decrypt a message on the basis of the ciphertext alone. Beyond that, the security of conventional encryption depends on the secrecy of the key, not the secrecy of the algorithm. That is, it is assumed that it is impractical to decrypt a message on the basis of the ciphertext *plus* knowledge of the encryption/decryption algorithm. In other words, we don't need to keep the algorithm secret; we need to keep only the key secret.

This feature of conventional encryption is what makes it feasible for widespread use. The fact that the algorithm need not be kept secret means that manufacturers can and have developed low-cost chip implementations of data encryption algorithms. These chips are widely available and incorporated into a number of products. With the use of conventional encryption, the principal security problem is maintaining the secrecy of the key.

Figure 1 takes a closer look at the essential elements of a conventional encryption scheme. There is some source for a message, which produces a message in plaintext, $X = [X_1, X_2, \ldots, X_M]$. The M elements of X are letters in some finite alphabet. Traditionally, the alphabet usually consisted of the 26 capital letters. Nowadays, the binary alphabet {0, 1} is typically used. For encryption, a key of the form $K = [K_1, K_2, \ldots, K_J]$ is generated.

If the key is generated at the message source, then it must also be provided to the destination by means of some secure channel. Alternatively, a third party could generate the key and securely deliver it to both source and destination.

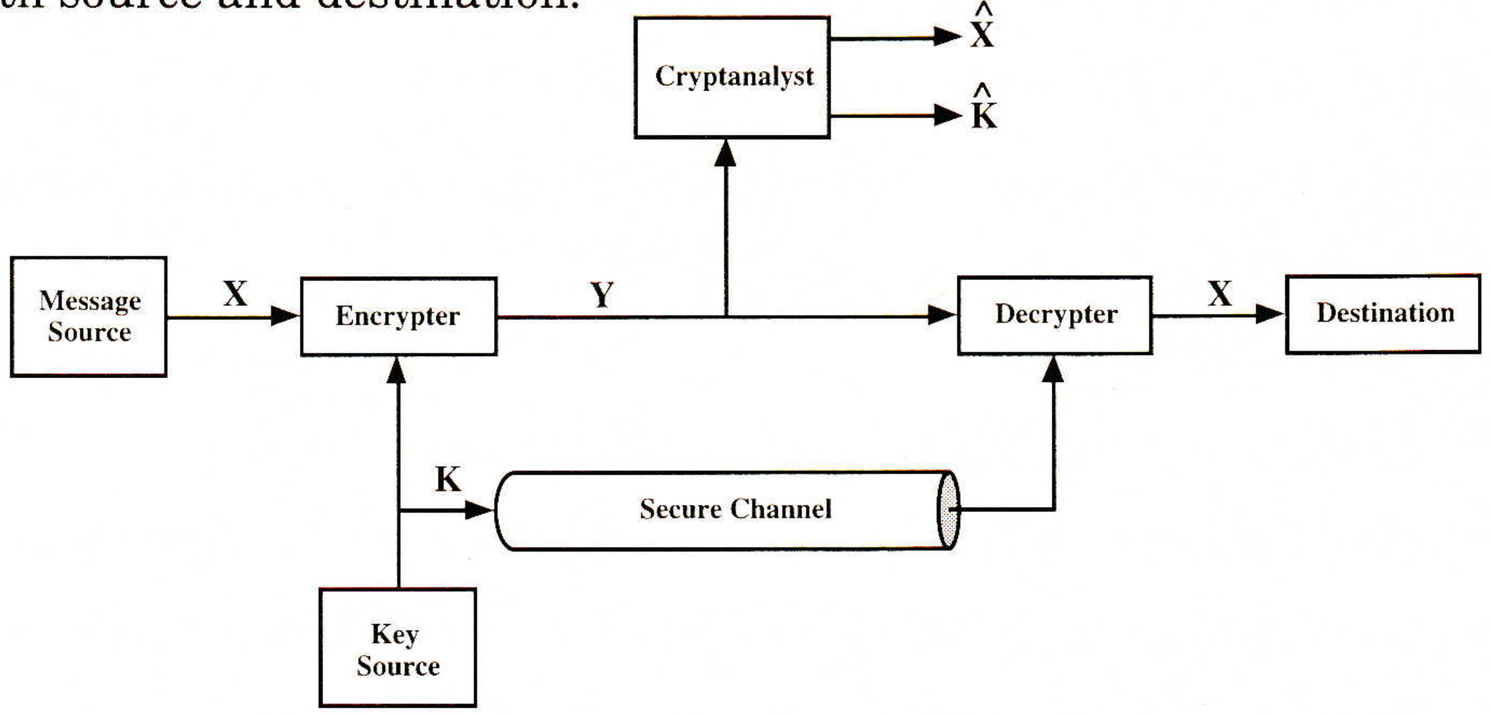


Figure 1: Model of conventional cryptosystem

With the message X and the key K as input, the encryption algorithm forms the ciphertext $Y = [Y_1, Y_2, \ldots, Y_N]$. We can write this as:

$$Y = E_K(X)$$

This notation indicates that Y is produced by using encryption algorithm E as a function of the plaintext X, with the specific function determined by the value of the key K.

The intended receiver, in possession of the key, is able to invert the transformation:

$$X = D_K(Y)$$

An opponent, observing Y but not having access to K or X, must attempt to recover X or K or both X and K. It is assumed that the opponent does have knowledge of the encryption (E) and decryption (D) algorithms. If the opponent is interested in only this particular message, then the focus of effort is to recover X by generating a plaintext estimate X'. Often, however, the opponent is interested in being able to read future messages as well, in which case an attempt is made to recover K by generating an estimate K'.

Block ciphers

The most commonly-used conventional encryption algorithms are *block ciphers*. A block cipher processes the plaintext input in fixed-size blocks, and produces a block of ciphertext of equal size for each plaintext block. Some of the most important block ciphers are listed below.

Algorithm	Key size (bits)	Block size (bits)	Example applications used in
DES	56	64	Kerberos, PEM, SNMPv2
Triple DES	112	64	PEM
IDEA	128	64	PGP
SKIPJACK	80	64	Clipper

DES = Data Encryption Standard
IDEA = International Data Encryption Algorithm
PEM = Privacy Enhanced Mail
SNMPv2 = Simple Network Management Protocol, version 2
PGP = Pretty Good Privacy

Table 1: Noteworthy conventional encryption algorithms

continued on next page

Cryptographic Algorithms *(continued)*

DES

The most widely used encryption scheme is defined in the *Data Encryption Standard* (DES) adopted in 1977 by the National Bureau of Standards, now the National Institute of Standards and Technology (NIST), as Federal Information Processing Standard 46 (FIPS PUB 46). In 1994, NIST "reaffirmed" DES for federal use for another 5 years [1]; NIST recommends the use of DES for applications other than the protection of classified information.

The overall scheme for DES encryption is illustrated in Figure 2. As with any encryption scheme, there are two inputs to the encryption function: the plaintext to be encrypted and the key. In this case, the plaintext must be 64 bits in length and the key is 56 bits in length.

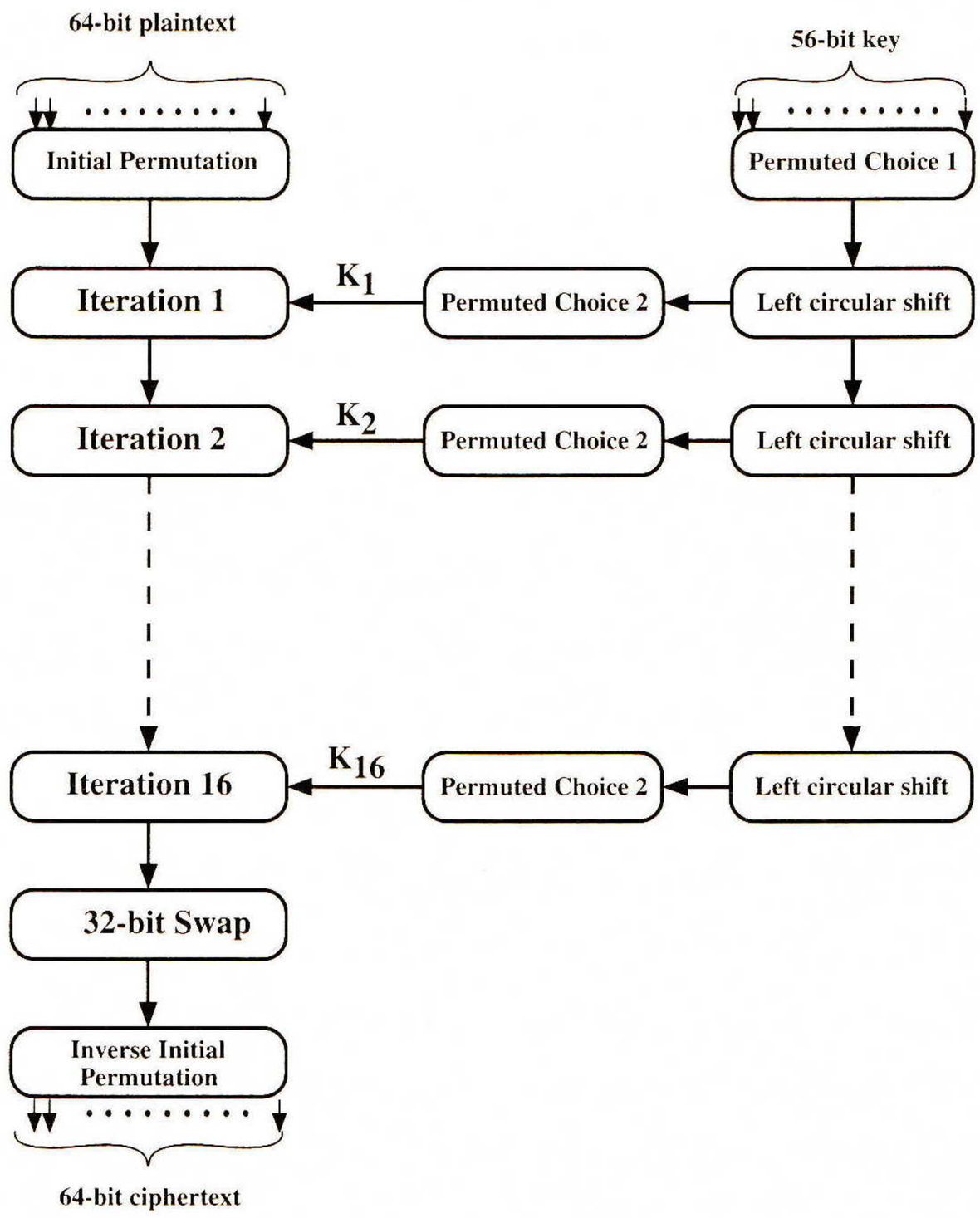


Figure 2: General depiction of DES Algorithm

Encryption

The processing of the plaintext proceeds in three phases. First, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the *permuted input*. This is followed by a phase consisting of 16 iterations of the same function. The output of the last (16th) iteration consists of 64 bits that are a function of the input plaintext and the key. The left and right halves of the output are swapped to produce the *preoutput*. Finally, the preoutput is passed through a permutation (IP^{-1}) that is the inverse of the initial permutation function, to produce the 64-bit ciphertext.

The right-hand portion of Figure 2 shows the way in which the 56-bit key is used. Initially, the key is passed through a permutation function. Then, for each of the 16 iterations, a *subkey* (K_i) is produced by the combination of a left circular shift and a permutation. The permutation function is the same for each iteration, but a different subkey is produced because of the repeated shifting of the key bits.

Figure 3 examines more closely the algorithm for a single iteration. The 64-bit permuted input passes through 16 iterations, producing an intermediate 64-bit value at the conclusion of each iteration. The left and right-half of each 64-bit intermediate value are treated as separate 32-bit quantities, labeled *L* (left) and *R* (right). The overall processing at each iteration can be summarized in the following formulas:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

where ⊕ denotes the bitwise XOR function.

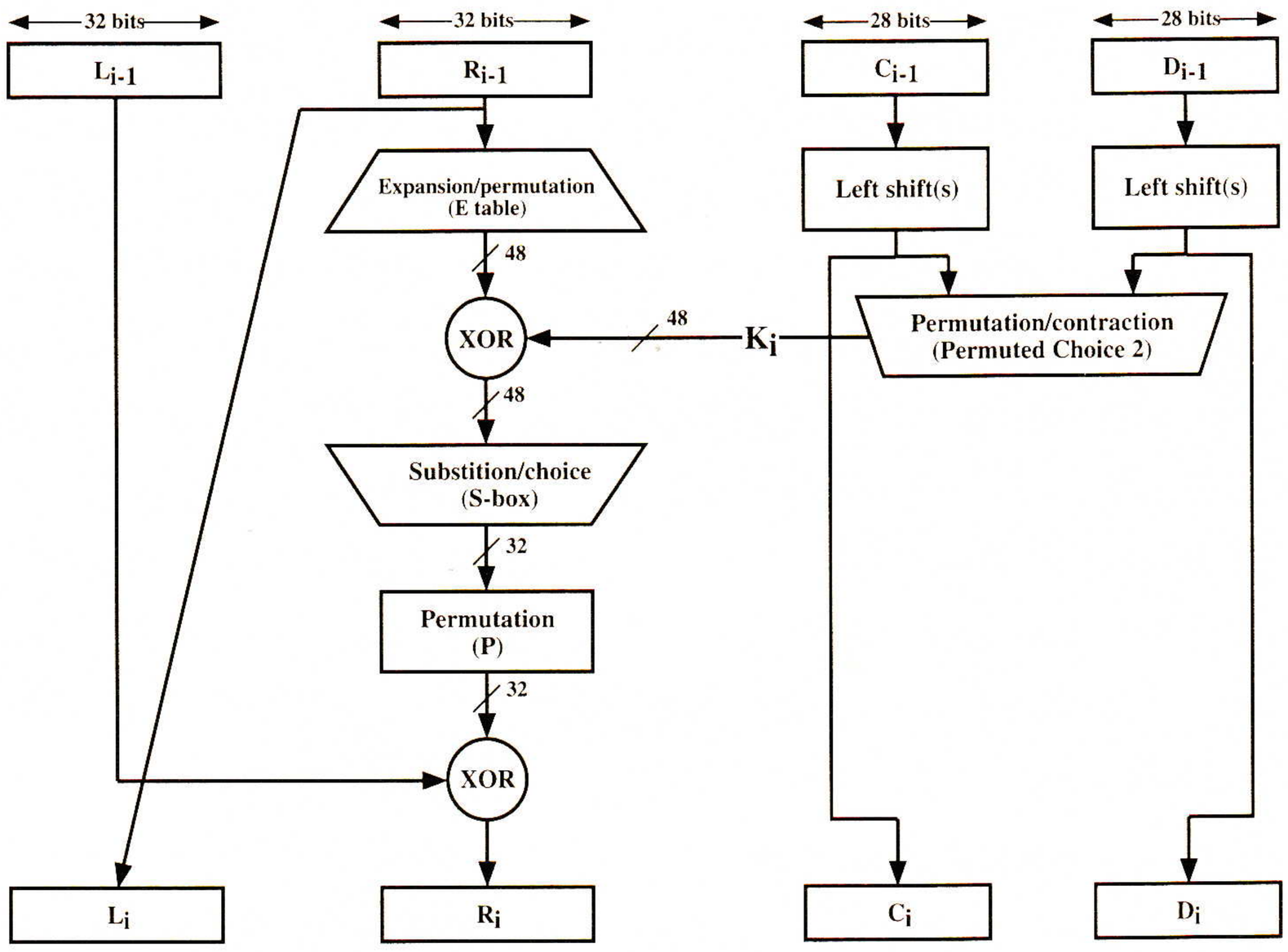


Figure 3: Single iteration of DES Algorithm

Thus, the left-hand output of an iteration (L_i) is simply equal to the right-hand input to that iteration (R_{i-1}). The right-hand output (R_i) is the exclusive-or of L_{i-1} and a complex function f of R_{i-1} and K_i. This complex function involves both permutation and substitution operations. The substitution operation, represented as tables called "S-boxes," simply maps each combination of 48 input bits into a particular 32-bit pattern.

Returning to Figure 3, we see that the 56-bit key used as input to the algorithm is first subjected to a permutation. The resulting 56-bit key is then treated as two 28-bit quantities, labeled C_0 and D_0. At each iteration, C and D are separately subjected to a circular left shift, or rotation, of 1 or 2 bits. These shifted values serve as input to the next iteration. They also serve as input to another permutation function, which produces a 48-bit output that serves as input to the function $f(R_{i-1}, K_i)$.

Decryption

The process of decryption with DES is essentially the same as the encryption process. The rule is as follows: Use the ciphertext as input to the DES algorithm, but use the keys K_i in reverse order. That is, use K_{16} on the first iteration, K_{15} on the second iteration, and so on until K_1 is used on the 16th and last iteration.

continued on next page

Cryptographic Algorithms *(continued)*

The strength of DES

Since its adoption as a federal standard, there have been lingering concerns about the level of security provided by DES. These concerns, by and large, fall into two areas: the nature of the algorithm and key size.

For many years, the more important concern was the possibility of exploiting the characteristics of the DES algorithm to perform cryptanalysis. The focus of concern has been on the eight substitution tables, or S-boxes, that are used in each iteration. Because the design criteria for these boxes, and indeed for the entire algorithm, have never been made public, there is a suspicion that the boxes were constructed in such a way that cryptanalysis is possible for an opponent who knows the weaknesses in the S-boxes. This assertion is tantalizing, and over the years a number of regularities and unexpected behaviors of the S-boxes have been discovered. Despite this, no one has so far succeeded in discovering the supposed fatal weaknesses in the S-boxes. Indeed, as advances in cryptanalytic techniques have occurred, the underlying strength of the DES algorithm has become more apparent. As of this writing, no practical attack method for DES has been published. Given that the algorithm has survived years of intensive scrutiny unscathed, it is probably safe to say that DES is one of the strongest encryption algorithms ever devised.

The more serious concern today, is the key size. With a key length of 56 bits, there are 2^{56} possible keys, which is approximately 7.6×10^{16} keys. Thus, on the face of it, a brute-force attack appears impractical. Assuming that on average half the key space has to be searched, a single machine performing one DES encryption per microsecond would take more than a thousand years to break the cipher.

However, the assumption of one encryption per microsecond is overly conservative. As far back as 1977, Diffie and Hellman, the inventors of public-key encryption, postulated that the technology existed to build a parallel machine with 1 million encryption devices, each of which could perform one encryption per microsecond [2]. The authors estimated that the cost would be about $20 million in 1977 dollars.

The most rigorous recent analysis of the problem was done by Wiener [3] and is based on a known plaintext attack. That is, it is assumed that the attacker has at least one (plaintext, ciphertext) pair. Wiener takes care to provide the details of his design. To quote his paper:

> "There have been numerous unverifiable claims about how fast the DES key space can be searched. To avoid adding to this list of questionable claims, a great deal of detail in the design of a key search machine is included in the appendices. This detailed work was done to obtain an accurate assessment of the cost of the machine and the time required to find a DES key. There are no plans to actually build such a machine."

Wiener reports on the design of a chip that uses pipelined techniques to achieve a key search rate of 50 million keys per second. Using 1993 costs, he designed a module that costs $100,000 and contains 5,760 key search chips. With this design, the following results are obtained:

Key Search Machine Unit Cost	Expected Search Time
$100,000	35 hours
$1,000,000	3.5 hours
$10,000,000	21 minutes

In addition, Wiener estimates a one-time development cost of about $500,000.

The Wiener design represents the culmination of years of concern about the security of DES and may in retrospect have been a turning point. As of the time of this writing, it still seems reasonable to rely on DES for personal and commercial applications. But the time has come to investigate alternatives for conventional encryption. Two of the most promising candidates for replacing DES are triple DES and IDEA.

Triple DES

Given the potential vulnerability of DES to a brute-force attack, there has been considerable interest in finding an alternative. One approach is to design a completely new algorithm, such as IDEA. An alternative, which would preserve the existing investment in software and equipment, is to use multiple encryption with DES and multiple keys.

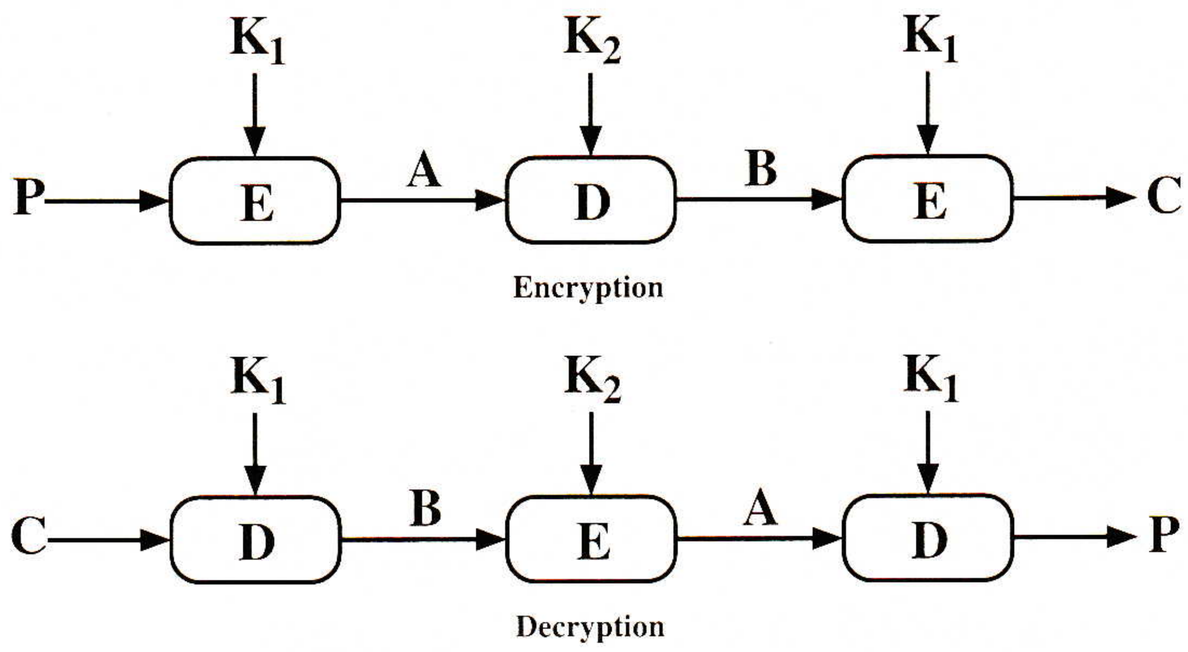


Figure 4: Triple DES

The most popular form of multiple DES, referred to as *Triple DES,* was first proposed by Tuchman [4] and first standardized in [5]. Triple DES uses two keys and three executions of the DES algorithm (Figure 4). The function follows an encrypt-decrypt-encrypt (EDE) sequence:

$$C = E_{K1} [D_{K2} [E_{K1} [P]]]$$

There is no cryptographic significance to the use of decryption for the second stage. Its only advantage is that it allows users of triple DES to decrypt data encrypted by users of the older single DES:

$$C = E_{K1} [D_{K1} [E_{K1} [P]]] = E_{K1} [P]$$

Although only two keys are used, three instances of the DES algorithm are required. It turns out that there is a simple technique, known as a "meet-in-the-middle attack," that would reduce a double DES system with 2 keys to the relative strength of ordinary single DES. With three iterations of the DES function, the effective key length is 112 bits.

IDEA

The *International Data Encryption Algorithm* (IDEA) is a block-oriented conventional encryption algorithm developed in 1990 by Xuejia Lai and James Massey of the Swiss Federal Institute of Technology. The original version was published in [6]. A revised version of the algorithm, designed to be stronger against recent advances in cryptanalytic attacks, was presented in [7].

The overall scheme for IDEA encryption is illustrated in Figure 5 on the following page. IDEA is a block cipher that uses a 128-bit key to encrypt data in blocks of 64 bits.

continued on next page

Cryptographic Algorithms *(continued)*

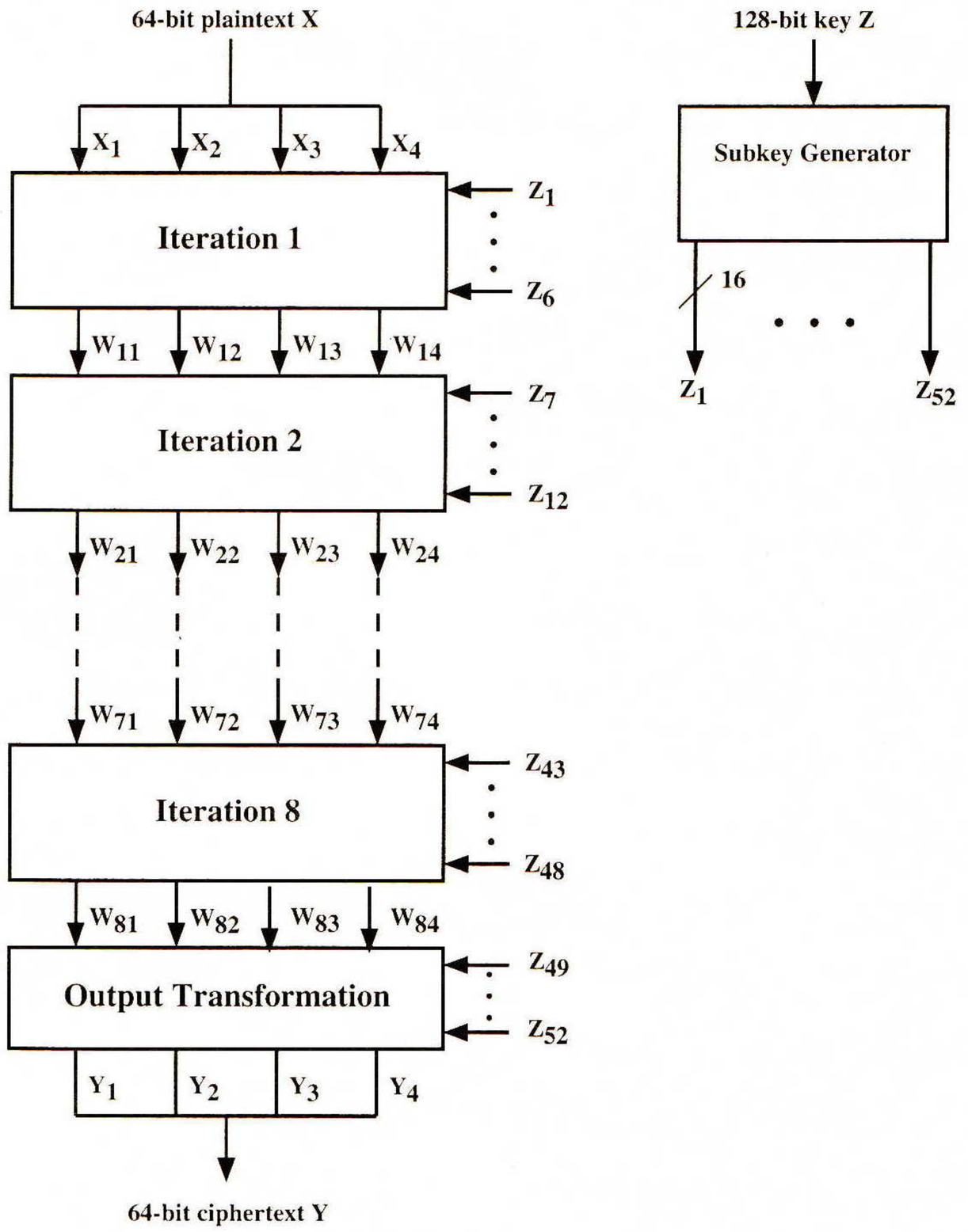


Figure 5: Overall IDEA structure

Iterations

The IDEA algorithm consists of eight rounds, or *iterations,* followed by a final transformation function. The algorithm breaks the input up into four 16-bit sub-blocks. Each of the iteration rounds takes four 16-bit sub-blocks as input and produces four 16-bit output blocks. The final transformation also produces four 16-bit blocks, which are concatenated to form the 64-bit ciphertext. Each of the iterations also makes use of six 16-bit subkeys, whereas the final transformation uses four subkeys, for a total of 52 subkeys. The right-hand portion of the figure indicates that these 52 subkeys are all generated from the original 128-bit key.

Each iteration of IDEA makes use of three different mathematical operations. Each operation is performed on two 16-bit inputs to produce a single 16-bit output. The operations are:

- Bit-by-bit exclusive-OR, denoted as $\oplus$.
- Addition of integers modulo 2^{16} (modulo 65536), with inputs and outputs treated as unsigned 16-bit integers. This operation is denoted as $\boxplus$
- Multiplication of integers modulo $2^{16} + 1$ (modulo 65537), with inputs and outputs treated as unsigned 16-bit integers, except that a block of all zeros is treated as representing 2^{16}. This operation is denoted as $\odot$

For example,

$$0000000000000000 \odot 1000000000000000 = 1000000000000001$$

because

$$2^{16} \times 2^{15} \bmod (2^{16} + 1) = 2^{15} + 1$$

These three operations are incompatible in the sense that:

1. No pair of the three operations satisfies a distributive law. For example:

$$a \boxplus (b \odot c) \neq (a \boxplus b) \odot (a \boxplus c)$$

2. No pair of the three operations satisfies an associative law. For example:

$$a \boxplus (b \oplus c) \neq (a \boxplus b) \oplus c$$

The use of these three separate operations in combination provides for a complex transformation of the input, making cryptanalysis much more difficult than with an algorithm such as DES, which relies solely on the XOR function.

Now let us look more closely at the algorithm for a single iteration, as illustrated in Figure 6. In fact, this figure shows the first iteration. Subsequent iterations have the same structure but with different subkey and plaintext-derived inputs.

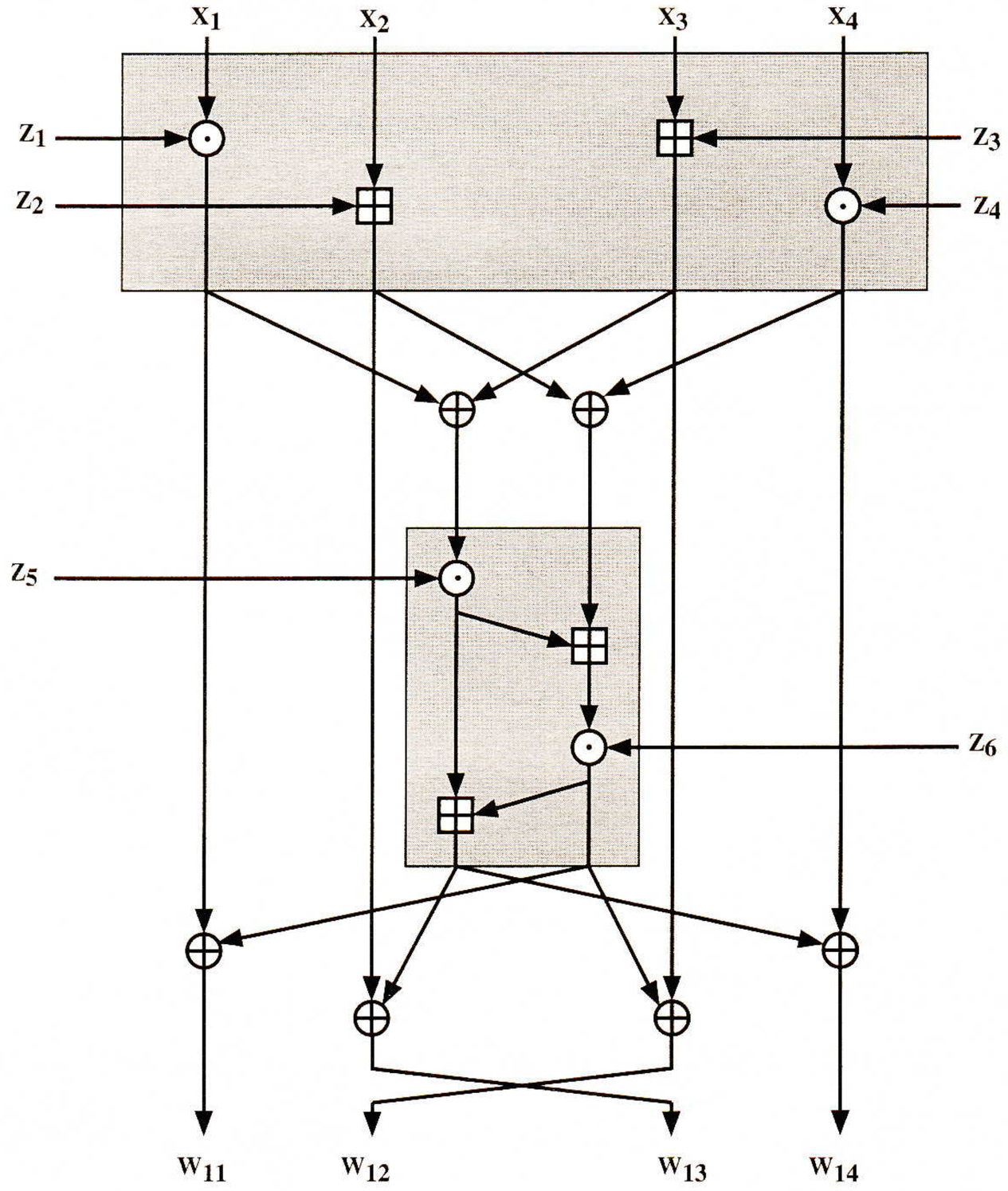


Figure 6: Single iteration of IDEA (first iteration)

continued on next page

Cryptographic Algorithms *(continued)*

The iteration begins with a transformation that combines the four input sub-blocks with four subkeys, using the addition and multiplication operations. This transformation is highlighted as the upper shaded rectangle. The four output blocks of this transformation are then combined using the XOR operation to form two 16-bit blocks that are input to the lower shaded rectangle, which also takes two subkeys as input and combines these inputs to produce two 16-bit outputs.

Finally, the four output blocks from the upper transformation are combined with the two output blocks of the MA structure using XOR to produce the four output blocks for this iteration. Note that the two outputs that are partially generated by the second and third inputs (X_2 and X_3) are interchanged to produce the second and third outputs (W_{12} and W_{13}). This increases the mixing of the bits being processed and makes the algorithm more resistant to cryptanalysis.

The ninth stage of the algorithm, labeled the output transformation stage in Figure 5, is has the same structure as the upper, shaded portion of the preceding iterations (Figure 6). The only difference is that the second and third inputs are interchanged before being applied to the operational units. In fact, this has the effect of undoing the interchange at the end of the eighth iteration. The reason for this extra interchange is so that decryption has the same structure as encryption, as will be seen. Note also that this ninth stage requires only four subkey inputs, compared to six subkey inputs for each of the first eight stages.

Finally, the subkeys for each iteration are generated by a series of shifts on the original 128-bit key.

IDEA appears to have a number of advantages over DES or even triple DES. The key length of 128 bits makes it more resistant to brute-force key search attacks. The internal structure of IDEA appears to be such that IDEA is more resistant to cryptanalysis than DES. Finally, IDEA was designed to facilitate both software and hardware implementations. For comparable implementations of IDEA and triple DES, IDEA should execute in much less time.

SKIPJACK

In April 1993, the Clinton administration announced a proposed encryption technology that, according to the announcement "will bring the Federal Government together with industry in a voluntary program to improve the security and privacy of telephone communication while meeting the legitimate needs of law enforcement." Subsequently, in July 1993, a more formal announcement appeared in the Federal Register as a request for comments on a proposed Federal Information Processing Standard. The overall approach was initially referred to as *Clipper* (because of a conflict with an existing trademark, the term Clipper is no longer used by the government), whereas the specific encryption algorithm is known as *SKIPJACK*.

The initiative, according to the government, has two objectives: (1) to encourage widespread use of an encryption technology that enable law enforcement agencies to continue, when lawfully authorized, to monitor and wiretap private communications, and (2) to provide the public with access to a sophisticated and powerful encryption technology to meet private needs, especially those of business, for maintaining confidentiality. According to the government, this and the preceding objective are considered equally important.

There have been widespread and vociferous negative reactions to this proposed program, together with supportive responses from a much smaller and less vocal number of individuals. For examples of pro and con, see [8] and [9], respectively.

The algorithm, referred to as SKIPJACK, is a conventional encryption algorithm. SKIPJACK is a block cipher that uses an 80-bit key (compared to 56 bits for DES and 128 bits for IDEA) to encrypt data in blocks of 64 bits. The algorithm involves 32 rounds of processing for a single encryption or decryption, using a complex nonlinear function for each round. That is about all that is publicly known about the algorithm.

Next Month: Part II: Public-key encryption and secure hash functions.

References

[1] National Institute of Standards and Technology (NIST), "Data Encryption Standard," FIPS PUB 46-2, June 1994.

[2] Diffie, W., and Hellman, M. "Exhaustive Cryptanalysis of the NBS Data Encryption Standard," *Computer,* June 1977.

[3] Wiener, M. "Efficient DES Key Search," *Proceedings, Crypto '93, 1993,* published by Springer-Verlag.

[4] Tuchman, W. "Hellman Presents No Shortcut Solutions to DES," *IEEE Spectrum,* July 1979.

[5] American National Standards Institute, *Financial Institution Key Management (Wholesale),* ANS X9.17, 1985.

[6] Lai, X., and Massey, J., "A Proposal for a New Block Encryption Standard," *Proceedings, EUROCRYPT '90,* 1990; published by Springer-Verlag.

[7] Lai, X., and Massey, J. "Markov Ciphers and Differential Cryptanalysis," *Proceedings, EUROCRYPT '91,* 1991; published by Springer-Verlag.

[8] Baker, S, "Why Clipper is Good for You," *Wired,* 2.06, June 1994.

[9] Barlow, J. "A Plain Text on Crypto Policy," *Communications of the ACM,* November 1993.

[10] Schiller, J., "Issues in Internet Security," *ConneXions,* Volume 7, No. 9, September 1993.

[11] *ConneXions,* Volume 4, No. 8, August 1990, "Special Issue on Network Management and Network Security."

[12] Schiller, J., "Kerberos: Network Authentication, *ConneXions,* Volume 4, No. 1, January 1990.

[13] Kaliski, B., "An Overview of Public-Key Cryptography Standards," *ConneXions,* Volume 6, No. 5, May 1992.

[This article is based on material in Bill Stallings' *Network and Internetwork Security,* ISBN 0-02- 415483-0 © 1994 by Prentice Hall. Used with permission. —Ed.]

WILLIAM STALLINGS is an independent consultant whose clients have included major corporations and government agencies in the United States and Europe. He is the author over over a dozen books on data communications and computers, including *Data and Computer Communications, Fourth Edition,* from Prentice-Hall. He is currently at work on *Protect Your Privacy: The User's Guide to PGP,* to be published by Prentice Hall in January. He holds a PhD from M.I.T. in Computer Science and a B.S. from Notre Dame in electrical engineering. Internet e-mail: `stallings@acm.org`

Security Awareness Increasing Within IETF

by Jim Galvin, Trusted Information Systems

Introduction

The *Internet Engineering Task Force* (IETF) is the protocol engineering, development, and standardization arm of the *Internet Architecture Board* (IAB). The IETF began in January 1986 as a forum for technical coordination by contractors for the then US Defense Advanced Projects Agency (DARPA), working on the ARPANET, US Defense Data Network (DDN), and the Internet core gateway system. Since then, the IETF has grown into a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet protocol architecture and the smooth operation of the Internet.

Technical activity on any specific topic in the IETF is addressed within *working groups,* which are organized roughly by function into nine technical *areas,* each led by one or more *area directors.*

The current director of the Security area is Jeff Schiller of MIT. The Security Area within the IETF is responsible for development of security-oriented protocols, security review of RFCs, development of candidate policies, and review of operational security on the Internet. Much of the work of the Security Area is performed in coordination with other IETF working groups.

The *Security Area Advisory Group* (SAAG) performs both consulting to other areas and direct management of working groups within the security area. Most of the SAAG's work consists of a set of formal work items corresponding to working groups within the IETF Security Area, security-relevant developments within other working groups, and internal SAAG work items that do not merit the creation of formal working groups but need some level of attention.

Three times a year, the IETF holds 4.5-day meetings comprising working group sessions, technical presentations, network status reports, working group reporting, and an open *Internet Engineering Steering Group* (IESG) meeting.

The 29th meeting of the IETF was held in Seattle, March 28–April 1, 1994. General Security awareness was at an all-time high at this meeting. Following is a summary of the SAAG meeting and a summary of the active working groups in the Security Area.

SAAG meeting summary

The Security Area Advisory Group met twice during this IETF session. Two areas felt to deserve special attention were one-time passwords and firewalls.

One-time passwords

It is obvious that allowing passwords to repeatedly appear in the clear on networks is a serious vulnerability. It is time to deprecate the usage of cleartext passwords and proactively assert the usage of one-time passwords. This conclusion represents a strategic direction of the Internet Security Area. There are at least three technologies available in this area: S/Key, Challenge Response Cards, and SecureID. Phil Karn (of Qualcomm) and Bill Simpson (of Computer Systems Consulting Services) agreed to draft a document describing S/Key. Glen Zorn (of Open Computing Security Group) agreed to draft a description of Digital Pathways, a challenge response card. It was also noted that the use of one-time passwords should become an integral part of direct access protocols, for example, TELNET, FTP, and PPP.

Firewalls

The use of firewalls in the Internet is becoming more popular as one means by which sites can protect themselves from attack. However, there is very little guidance as to how to best make use of a firewall, in particular where and how a firewall fits in a security architecture. It was noted that John Wack is the point of contact at NIST for a document being prepared that may discuss some of these issues.

Security Area working group summary

Included below is a summary of the active working groups in the security area. For each working group the name and e-mail address of the Chair is provided.

Authorization and Access Control (AAC)

Chair: Clifford Neuman <bcn@isi.edu>

The goal of the *Authorization and Access Control* (AAC) Working Group is to develop guidelines and an Application Program Interface (API) through which network-accessible applications can uniformly specify access control information. This API will allow applications to make access control decisions when clients are not local users, might not be members of a common organization, and often not known to the service or application in advance. The authorization and access control draft that exists within the working group will be revised and reviewed by the working group in Toronto (July 1994) before being published as an informational document. At that time the charter of this working group will be revised.

Common Authentication Technology (CAT)

Chair: John Linn <linn@security.ov.com>

The goal of the *Common Authentication Technology* (CAT) Working Group is to provide strong authentication to a variety of protocol callers in a manner that insulates those callers from the specifics of underlying security mechanisms. By separating security implementation tasks from the tasks of integrating security data elements into caller protocols, those tasks can be partitioned and performed separately by implementors with different areas of expertise. This provides leverage for the IETF community's security-oriented resources and allows protocol implementors to focus on the functions their protocols are designed to provide rather than on characteristics of security mechanisms.

CAT seeks to encourage uniformity and modularity in security approaches, supporting the use of common techniques and accommodating evolution of underlying technologies.

The status of ongoing GSS-API and application development and testing was reviewed; there are now two independent Kerberos V5 GSS-API implementations. Various demonstration applications have been implemented in order to validate interoperability and in preparation for advancing RFCs 1508–1510 to Draft Standard Status. The group believes that the FTP Security Internet-Draft is ready for advancement to Proposed Standard.

IP Security Protocol (IPSEC)

Chairs: James Zmuda <zmuda@mls.hac.com>
Paul Lambert <paul_lambert@email.mot.com>

Rapid advances in communication technology have accentuated the need for security in the Internet. The *IP Security Protocol* (IPSEC) Working Group will develop mechanisms to protect client protocols of IP. A security protocol in the network layer will be developed to provide cryptographic security services that will flexibly support combinations of authentication, integrity, access control, and confidentiality.

continued on next page

Security Awareness Within IETF *(continued)*

The protocol formats for the *IP Security Protocol* (IPSP) will be independent of the cryptographic algorithm. The preliminary goals will specifically pursue host-to-host security followed by subnet-to-subnet and host-to-subnet topologies.

This working group has been slow in developing an Internet Draft for the IP security protocol. There are many specifications, and there is no consensus as to which approach is preferred. In addition, there are also many implementations of the various specifications, none of which interoperate. An interoperability demonstration was scheduled for the Toronto (July 1994) meeting. The group did agree on the use of the DES and MD5 algorithms and manual key management for the initial interoperability tests.

Network Access Server Requirements (NASREQ)

Chairs: Allan Rubens <acr@merit.edu>;
John Vollbrecht <jrv@merit.edu>

The *Network Access Server Requirements* Working Group has as its primary goal to identify functions and services that should be present in IP Network Access Servers (NASs) and to specify the standards that provide for these functions and services. The term "Network Access Server" is used instead of the more conventional term "Terminal Server" as it more accurately describes the functions of interest to this group. A "Network Access Server" is a device that provides for the attachment of both traditional "dumb terminals" and terminal emulators as well as workstations, PCs, or routers utilizing a serial line framing protocol such as PPP or SLIP. An authentication and authorization model document has been distributed. When it has been revised and submitted for publication as an informational document, this working group will be discontinued.

Privacy Enhanced Mail (PEM)

Chair: Stephen Kent <kent@bbn.com>

Privacy-Enhanced Mail (PEM) is the outgrowth of work by the *Privacy and Security Research Group* (PSRG) of the Internet Research Task Force (IRTF). At the heart of PEM is a set of procedures for transforming RFC 822 messages in such a fashion as to provide integrity, data origin authenticity, and, optionally, confidentiality. PEM may be employed with either symmetric or asymmetric cryptographic key distribution mechanisms. Because the asymmetric (public-key) mechanisms are better suited to the large scale, heterogeneously administered environment characteristic of the Internet, to date only those mechanisms have been standardized. The standard form adopted by PEM is largely a profile of the CCITT X.509 (Directory Authentication Framework) recommendation. PEM is defined by a series of documents. The first in the series defines the message processing procedures. The second defines the public-key certification system adopted for use with PEM. The third provides definitions and identifiers for various algorithms used by PEM. The fourth defines message formats and conventions for user registration and *Certificate Revocation List* (CRL) distribution.

The PEM working group is being reorganized. Specifically, it will focus its work on completing the current PEM-MIME specification. New working groups (as yet not chartered) will form to discuss various possible methods of distributing keys, for use with both PEM-MIME and other protocols that can benefit from a key distribution infrastructure. The principal topic of discussion at this meeting was the PEM and MIME integration document, which was distributed and reviewed. Some minor revisions are required.

Domain Name System Security (DNSSEC)

Chair: James M. Galvin <galvin@tis.com>

The *Domain Name System* (DNS) Security Working Group (dnssec) will specify enhancements to the DNS protocol to protect the DNS against unauthorized modification of data and against masquerading of DNS data origin. That is, it will add data integrity and authentication capabilities to the DNS. The specific mechanism to be added to the DNS protocol will be a digital signature. The proposed security enhancements drafted by Donald Eastlake and Charlie Kaufman were reviewed. The desired requirements specified at the Houston meeting were reviewed, followed by a presentation and discussion of the proposal.

Further information

For further information about the Security Area Advisory Group (SAAG), send a request to: **saag@tis.com.**

To join the IETF general discussion list, send an e-mail request to: **ietf-request@cnri.reston.va.us.** To join other mailing lists, send a request to the associated request list. All internet mailing lists have a companion "**-request**" list. Send requests to join a list to: **<listname>-request@<listhost>**.

Information and logistics about upcoming meetings of the IETF are distributed on the IETF announcement mailing list. For general inquiries about the IETF, e-mail: **ietf-info@cnri.reston.va.us.** An archive of mail sent to the IETF list is available for anonymous FTP from the directory **/ietf-mail-archive/ietf** on Internet host **cnri.reston.va.us.**

A Proceedings of each IETF plenary is published, which includes reports from each area, each working group, and each Technical Presentation.

[Ed.: As you can tell from the above, there is a great deal of activity within the IETF related to security. Thus, this article should be considered a "snapshot" taken in the late spring of 1994. Dr. Galvin has promised to send us updates on the IETF security activities from time to time. A version of this article appeared in the *Data Security Letter,* No. 48, April 1994. For more information contact: **dsl@tis.com**].

References

[1] Dern, D., "Interview with Steve Kent on Internet Security," *ConneXions,* Volume 4, No. 2, February 1990.

[2] Galvin, J., "The Deployment of Privacy Enhanced Mail," *ConneXions,* Volume 5, No. 10, October 1991.

[3] Galvin, J., "Components of OSI: The Security Architecture," *ConneXions,* Volume 4, No. 8, August 1990.

[4] Schiller, J., "Issues in Internet Security," *ConneXions,* Volume 7, No. 9, September 1993.

[5] *ConneXions,* Volume 4, No. 8, August 1990, "Special Issue on Network Management and Network Security."

JAMES M. GALVIN is a Senior Computer Scientist at Trusted Information Systems (TIS). Dr. Galvin's responsibilities emphasize communications security, especially computer networks, architectures, policies, and procedures. He is a principal in the development of TIS' openly available implementation of Privacy Enhanced Mail. He is very active in the IETF Security Area, serving as Executive Director of the Security Directorate, and past Chair of the OSI Implementor's Workshop Security Special Interest Group, hosted quarterly by the National Institute of Standards and Technology. He received his Ph.D. and M.S. degrees, both in Computer Science, from the University of Delaware in 1988 and 1986, respectively. In 1982, he received his B.S. in Computer Science and Mathematics from Moravian College in Bethlehem, PA. E-mail: **galvin@tis.com**

Windows Sockets 2.0

The Next Generation Transport API for Microsoft Windows

by J. Allard, Microsoft Corporation

Overview

The Windows Sockets API 1.1 is a sockets-style transport interface for the Microsoft® Windows™ family of operating systems. Originally inspired by the need for a binary-compatible interface to multiple-vendors's TCP/IP stacks under Windows, a great deal of momentum has built around the API in both the commercial and public domain development communities. Over 100 Windows Sockets-compatible applications have been released to date. The widespread popularity of Windows Sockets has led to the desire to extend the API to facilitate the next generation of network-aware applications as well as to embrace new technologies such as IPng. This article serves as an overview of the scope of the Windows Sockets 2.0 effort and a call for your participation as an application developer, transport provider, or end-user.

Mission statement

The Windows Sockets 2.0 effort is chartered to design a ubiquitous transport-level API for the Microsoft Windows family of systems products. The API will be designed to facilitate transparent access to multiple transport families and permit third party transport vendors to work into the scheme. In order to meet our goal of producing a specification by April 30, 1995, a pragmatic philosophy will be required. The basic ground rules include:

- Backward compatibility with Windows Sockets 1.1
- Limiting changes to the Windows Sockets 1.1 specification to clarifications only
- New functionality will be clearly distinguishable from the 1.1 API
- Performance efficiency consideration
- Optional functionality should be generally useful and the exception, rather than the rule

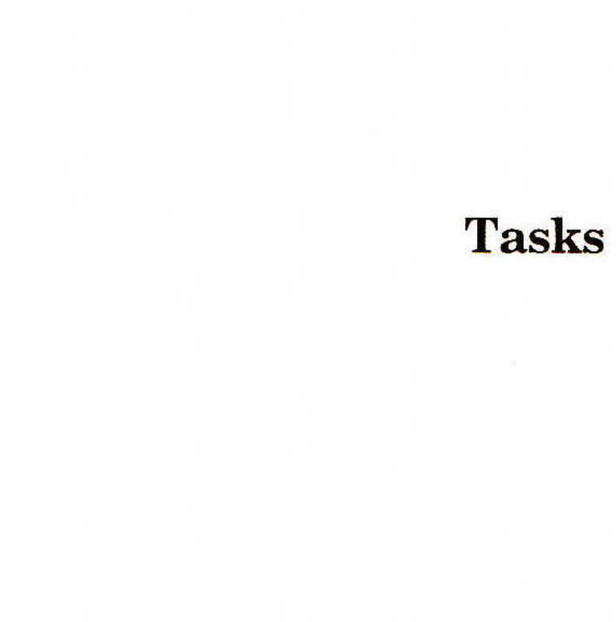
Tasks

The fundamental tasks of the effort fall into three specific areas:

API Extensions: Windows Sockets 1.1 has generated a great deal of momentum, resulting in a number of suggested improvements to the existing specification. Although some of these extensions will undoubtedly be transport-related, the primary focus will be in adding useful functionality which has been proposed by application vendors. Some examples in this area include improved support for non-**C** languages (e.g., BASIC, Pascal, **C++**), and transaction-oriented protocol support. As some of these extensions will be made optional, a mechanism for vendors and applications to deal with optional APIs will be required.

Multi-transport Capabilities: Although Windows Sockets 1.1 permits the ability to address multiple transport families, two problems persist in the effective implementation of transport-independent applications. The primary issue is that applications must be aware of the addressing format of the transports below it, with no simple mechanism to enumerate the transports available on the local host.

OS Considerations and Architecture: The strong momentum behind Windows Sockets, and the motivation to support multiple transports requires architectural considerations in multi-vendor environments. It will be desirable for an application to use multiple transports, provided from different vendors, on the same system without conflict.



Organization

Given the scale of the Windows Sockets 2.0 effort, a more granular organization will be required. Based on our experience with the 1.1 effort, and the proposals which have been presented, an organization will be formed with specific groups to address specific issues. This organization is designed to facilitate focus, parallel development of the specification during the design process.

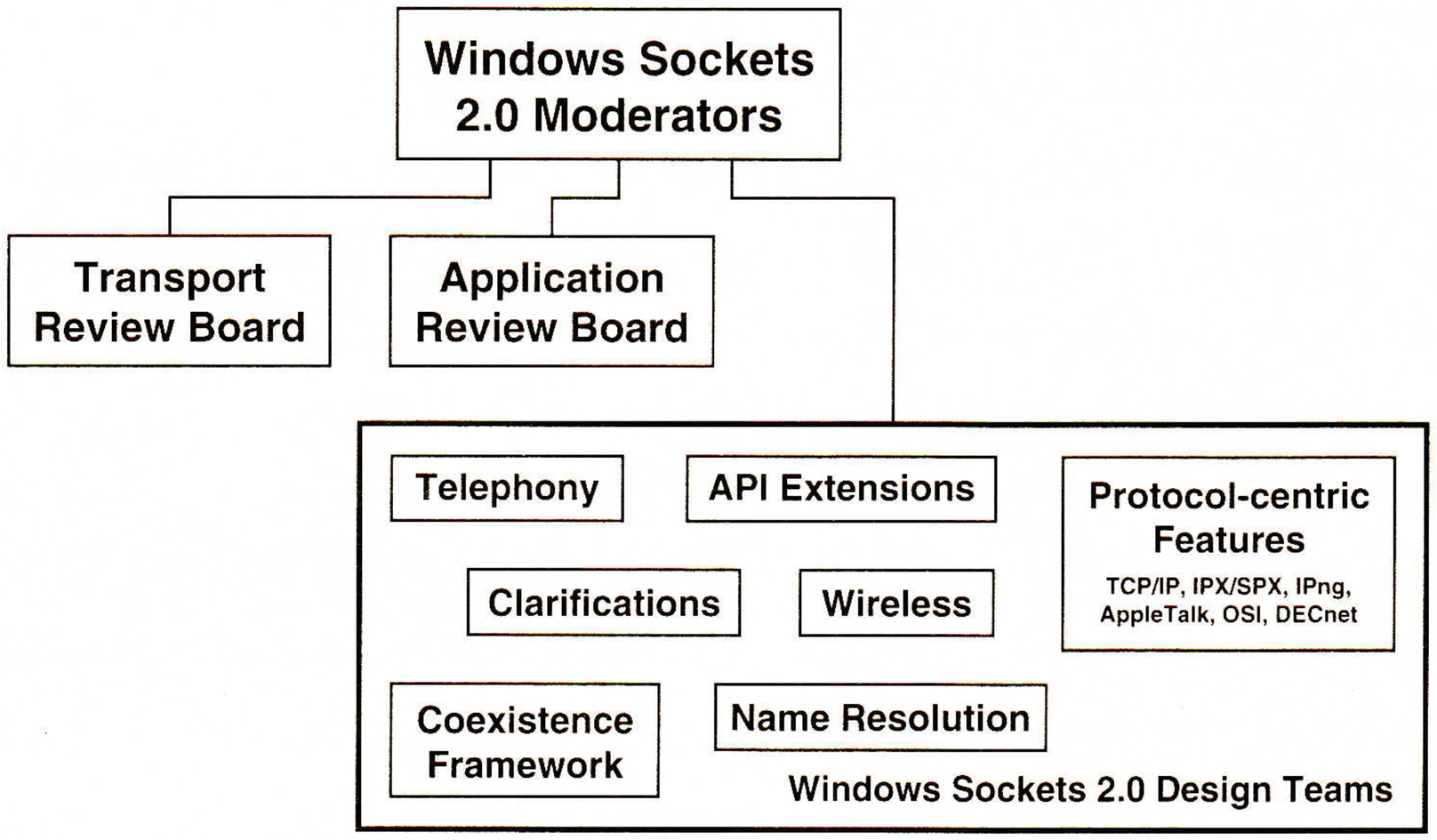


Figure 1: The Windows Sockets 2.0 Organization

Design Teams

A number of design teams will be established to facilitate a "divide-and-conquer" approach to the large number of technical issues facing the Windows Sockets 2.0 effort. Each group will have an administrator responsible for coordination of the design discussions. The following design teams are proposed at this time:

Team	Task
Generic API Extensions:	To specify general extensions generally applicable to multiple transport families
Specification Clarifications:	Resolve ambiguities in the existing specification
Name Resolution:	Design of a transport-independent, and directory service independent fashion
Coexistence Framework:	Ensure that vendors can implement and applications can participate on systems with multi-vendor transport support
Telephony:	Extensions to Windows Sockets for ISDN, ATM, etc..
Wireless:	Wireless and mobile-centric extensions
Protocol Design Teams:	Teams responsible for transport-specific Windows Sockets extensions for TCP/IP, IPng, IPX/SPX, AppleTalk, OSI, and DECnet

continued on next page

Windows Sockets 2.0 *(continued)*

Review Boards

Two review boards will be chartered to "steer" the efforts of the design teams—an application board and a transport board. The application review board will review submissions from each of the design teams, with particular attention paid to ease-of-use, consistency, and general usefulness of the proposals. The transport review board will focus mainly on implementation and performance issues in addition to applicability and affect of the new APIs on different transports. To limit the size of the boards, a maximum of one representative from each organization may participate in each review board.

An administrator for each review board will be selected to provide regular status updates to the moderators and to facilitate periodic discussions (both electronic as well as the occasional voice conferences) between the review board members.

Moderators

The Windows Sockets 2.0 moderators will largely act as coordinators of the effort. Specifically, the moderators will be responsible for coordination between the design teams, implementation consistency, header file consistency, and public relations. Martin Hall of JSB Corporation, Mark Towfiq of SunSelect, and David Treadwell of Microsoft Corporation will act in this capacity.

Schedule

The following schedule is proposed:

Functionality group requirements drafts	June 6, 1994
Final functionality group requirements	June 30, 1994
Draft functionality group specifications	August 31, 1994
Final draft functionality group specifications	November 30, 1994
Preliminary interpretability testing	January 1995
Functionality specification clarifications	February 28, 1995
Interoperability testing	March 1995
Completed Windows Sockets 2.0 specification	April 30, 1995

Mailing lists

A number of new, Windows Sockets 2.0 specific mailing lists have been set up by Intel Corp. and are active for participation:

`winsock-2`

`ws2-generic-api-ext`

`ws2-spec-clarif`

`ws2-name-resolution`

`ws2-oper-framework`

`ws2-tcp-ip`

`ws2-ipx-spx`

`ws2-appletalk`

`ws2-conn-oriented-media` ; (formerly `telephony`)

`ws2-osi`

`ws2-wireless`

Subscriptions to these mailing lists are open to any interested parties.

Subscription instructions are given below:

- Address all Windows Sockets 2.0-related e-mail administration requests to: `majordomo@mailbag.intel.com`
- Unlike some listservs, all text in the subject line is ignored
- All list commands must appear in the e-mail body, one command per line
- In the command summaries below, `<list>` is to be replaced with the full name of a particular list as shown above.
- Available commands:

`lists`	; provides a list of all mailing lists served by this system
`subscribe <list>`	; subscribe to the indicated list
`unsubscribe <list>`	; unsubscribe from the indicated list
`which`	; find out which lists you are currently subscribed to
`info <list>`	; get a brief info file for the indicated list
`help`	; get the rest of the available commands and options

- Post messages to a list by simply addressing your e-mail to: `<list>@mailbag.intel.com`

More information

Microsoft has volunteered to host Windows Sockets 2.0 information on their *World Wide Web* server: `www.microsoft.com`

References

[1] "Windows Sockets: Where necessity is the mother of *re*-invention," Martin Hall, *ConneXions,* Volume 7, No. 9, September 1993.

[2] "Windows Sockets: An Open Interface for Network Programming under Microsoft Windows," Martin Hall, Mark Towfiq, Geoff Arnold, David Treadwell, Henry Sanders, January 20 1993.

[3] "A Guide to Windows Sockets," Martin Hall, June 1 1993.

[4] "Plugging into TCP/IP with Windows Sockets," Victor Volkman *Windows/DOS Developers Journal,* Volume 3, No. 12, December 1992.

[5] "Untangling the Windows Sockets API," Mike Calbaum, Frank Porcaro, Mark Ruegsegger, Bruce Backman, *Dr. Dobbs Journal,* #197 February 1993.

[6] "The Windows Sockets API," Ralph Davis, Chapter 6 in *Windows Network Programming,* from "The Andrew Schulman Programming Series," Addison-Wesley Publishing Company, 1993.

[7] "Windows Sockets—Get Plugged in to Serious Network Programming," J. Allard, Keith Moore, David Treadwell, *Microsoft Systems Journal,* July 1993.

J. ALLARD is the Program Manager for Internet Technologies at Microsoft. His team focuses on building powerful and easy-to-use internetworking technologies for the Microsoft Windows operating systems family. J. is an active member of the IETF, currently serving on the IPng directorate, and singing the praise of variable length addresses. His trademark `.signature` quote reads "On the Internet, nobody knows you're running Windows NT." E-mail: `jallard@microsoft.com`

Corporate IP—An Orthogonal View

by George Abe, Cisco Systems

Introduction

Commercial IP services have become a familiar part of the Internet landscape. Members of the *Commercial Internet eXchange* (CIX) have provided IP connectivity and applications services to the commercial community for years now. But another type of service, called herein *Corporate IP,* has also emerged, as an option for corporate clients who wish to connect LANs. Corporate IP is different from Commercial IP. They differ as to their origins, services and markets. Corporate users may wish to investigate how the services of Corporate IP providers can supplement the services they receive from Commercial IP providers or public carriers.

The origins of Corporate IP

Corporate IP providers typically began as public carriers or value added networks who were accustomed to providing shared X.25 or circuit switching services to corporate clients. Often the corporate clients had global operational requirements and needed support for mission critical applications, such as transactions systems. Examples of such providers are Infonet and Sprint.

These companies had the business infrastructure for dealing with global corporate users. They had sales and marketing organizations, invoicing and customer tracking systems, service and support organizations and other elements of business practices, not the least of which is mind set. They were also accustomed to developing customer requirements and delivering turnkey, customized solutions. Their weak suit was technical innovation, but often their customers were more interested in scaling, stability, cost and service. Solid, boring, cheap.

On the other hand, commercial IP providers began as regional or not-for-profit organizations offering public IP services to government, educational and research institutions. These organizations were accustomed to offering IP services and applications (e-mail, directory services, etc.). The strong suit of the Internet providers was (and still is) technical innovation. But they did not have the business infrastructure to manage a for-profit enterprise for growth.

For the Infonets and Sprints of the world, it was necessary to add IP knowledge to sell commercial IP. For the Internet providers, it was necessary to add business infrastructure (paperwork, boring details) to sell commercial IP.

Requirements

What service requirements make Corporate IP different from Commercial IP?

- *Virtual Private Networking / Dedicated Resources:* Corporate IP serves primarily an *intra*-organizational requirement whereas Commercial providers served primarily an *inter*-organizational requirement. For corporate users, something like the 80/20 rule applies. Eighty percent of corporate traffic (if not more) stays within the company. Even when dial services are available, they usually want calls restricted to within the company. This imposes strong access control requirements on the service provider.

- *Support for Protocols Beyond IP:* Corporations uses other protocols than IP. Notable among these are SNA, IPX and AppleTalk In Europe, X.25 is important. Corporate IP providers must accommodate these protocols, typically with tunneling, but occasionally with native support. Infonet, for example, will route CLNP natively. Both Infonet and Sprint also have X.25 services.

• *Connection to Internal, Corporate Networks:* Large corporate users usually have internal, private networks. This can be an X.25, IP, TDM or switched VPN. It is the role of the shared corporate service to extend coverage, provide redundancy or replace parts of the private infrastructure. It is necessary to have familiarity with the network-to-network interfaces (X.75, NNI, BGP) to connect to internal networks.

• *Users Less Knowledgeable About IP:* Corporate users are usually less knowledgeable about IP than users of commercial, regional IP networks. Often the networking decision makers are from telecoms (voice) management or have prior SNA or X.25 experience. This imposes a strong support requirement on the service provider, both on the up-front sale and the continuing relationship.

• *Industrial Strength Support:* Because users are less knowledgeable and applications are often mission critical, the support requirements of Corporate IP providers is very high. Industrial strength support consists of end-to-end provisioning of equipment, global availability of telephone support (often with a local or toll free call), round the clock support, multilingual support and service warranties.

• *Customer premises hardware:* Virtual private services are (or should be) turnkey offerings. The service provider should provide all equipment needed on the customer premises as an option. The service provider will configure customer premises equipment whether or not it is obtained from the service provider as a normal option. It is important that the provider offer hardware maintenance and sparing services.

• *Global coverage:* Commercial IP services often have limited geographical scope. This will not do for big companies who need overseas service as well.

• *7 by 24 by 365, multilingual support:* Help desks (accessible by local calling or toll free numbers) must be staffed around the clock because global users know no time zones. Further it is useful, if only for marketing and public relations reasons, to have multilingual help desks.

• *Service Level Agreements (warranties):* The service business is becoming increasingly competitive for IP. Large carriers are being attracted to this market, and they have a background in service level agreements (SLAs). SLAs offer restitution to the end user for episodes of very bad service. Proper metrics and contractual terms are needed to offer this.

• *Who can I sue?* Many customers like to deal with large service providers for mission critical services. If everything goes to Hades in a handbasket, life can be made truly uncomfortable for the errant provider.

Conclusion

The commercial IP providers are improving their services in all these areas. But most have determined that it is probably not wise to take on Corporate IP providers head on, especially as more public carriers begin to offer IP services. The areas in which commercial IP providers can retain an edge are (1) intercorporate services, such as e-mail and database access, and (2) technically innovative services, such as PSI's cable service.

Corporate users should consider a mix of Commercial and Corporate IP services when addressing the full range of intracorporate and intercorporate user requirements.

GEORGE ABE holds an A.B. and M.S. from UCLA. He currently is on the Business Development staff at Cisco Systems. Prior to that he was Director of Program Management at Infonet Services Corporation. E-mail: gabe@diablo.cisco.com

DMTF: A Foundation for Systems Management?

by John McConnell, McConnell Consulting

Introduction

Managing distributed computing systems has been an unrewarding, demanding task. System management problems have become more intractable with the introduction of client–server architectures throughout organizations. This article looks at DMTF as an emerging systems management framework; will it deliver viable solutions or become another failed attempt by a consortium of vendors?

Historical factors have contributed to the difficulties of systems management—each operating system has its own management tools with different levels of maturity, leading to multiple proprietary solutions. Many of the available products are also highly fragmented: for example, one software distribution package may also incorporate asset and inventory management, while another package distributes software and does virus checking, while yet a third does asset and inventory management exclusively. Poor modularity and lack of interoperability have led to a multitude of tools making the administrator's job quite difficult in the heterogeneous environment we must manage.

Another contributing factor is the relative immaturity of systems management tools relative to LAN or WAN network management solutions. Network management has been an issue for a substantially longer period of time and more sophisticated tools have been deployed. However, the systems management area need not take as long to reach the same level of maturity since many of the technologies may be transferable from network management.

For example, intelligent network management applications analyze the flows between different internetworked segments and then recommend actions to optimize performance. The same types of pattern recognition and analysis can be applied to interactions between sets of clients and servers. The systems administrator is still struggling to bring this chaotic distributed environment under better control.

The need for better systems management tools is made more imperative by the increasing rates of growth of client–server networks. Today's administrators cannot cope with this growth by scaling their management staffs appropriately. Skilled administrators are scarce and expensive and they must be leveraged with better systems management solutions.

DMTF: Another approach

The *Desktop Management Task Force* (DMTF) was formed in May 1992 by a consortium of eight leading vendors. The founders were: IBM, Hewlett-Packard, Digital Equipment Corporation, Intel, SynOptics, Microsoft and Novell—an influential group. For example, IBM, Hewlett-Packard, Digital and Sun are all strong players in the UNIX workstation marketplace. Novell and Microsoft are the industry's two leading operating systems vendors. SynOptics, an intelligent hub vendor, is also establishing a strong presence as a management solutions provider. Finally, Intel is one of the largest makers of computer chips and can be expected to embed management solutions in their silicon in the future.

The DMTF was formed to expedite the delivery of open desktop management interfaces so that vendors and application providers could focus their resources on the building manageable components and applications.

The DMTF has several design goals, including:

- Independence of any specific computer or operating system
- Easy vendor adoption through development of Beta code, software developer kits, implementers workshops, and joint marketing efforts
- Independence of any network or systems management protocols with
- Mapping of information from DMTF to current management protocols such as SNMP and CMIP.

DMTF was also designed to be usable locally so that unattached users would also have a simpler job managing their desktops. The real power of DMTF is the possibility of a standard, open framework for managing diverse systems through the attached network.

Basic architecture

DMTF architecture is structurally simple in order to meet the goal of easy implementation. The *Desktop Management Interface* (DMI) contains three major parts: the *Management Interface* (MI), the *Service Layer* (SL), and the *Component Interface* (CI). The two interfaces insulate manageable components and management applications from any of the specific details of the operating system or hardware environment in which they run.

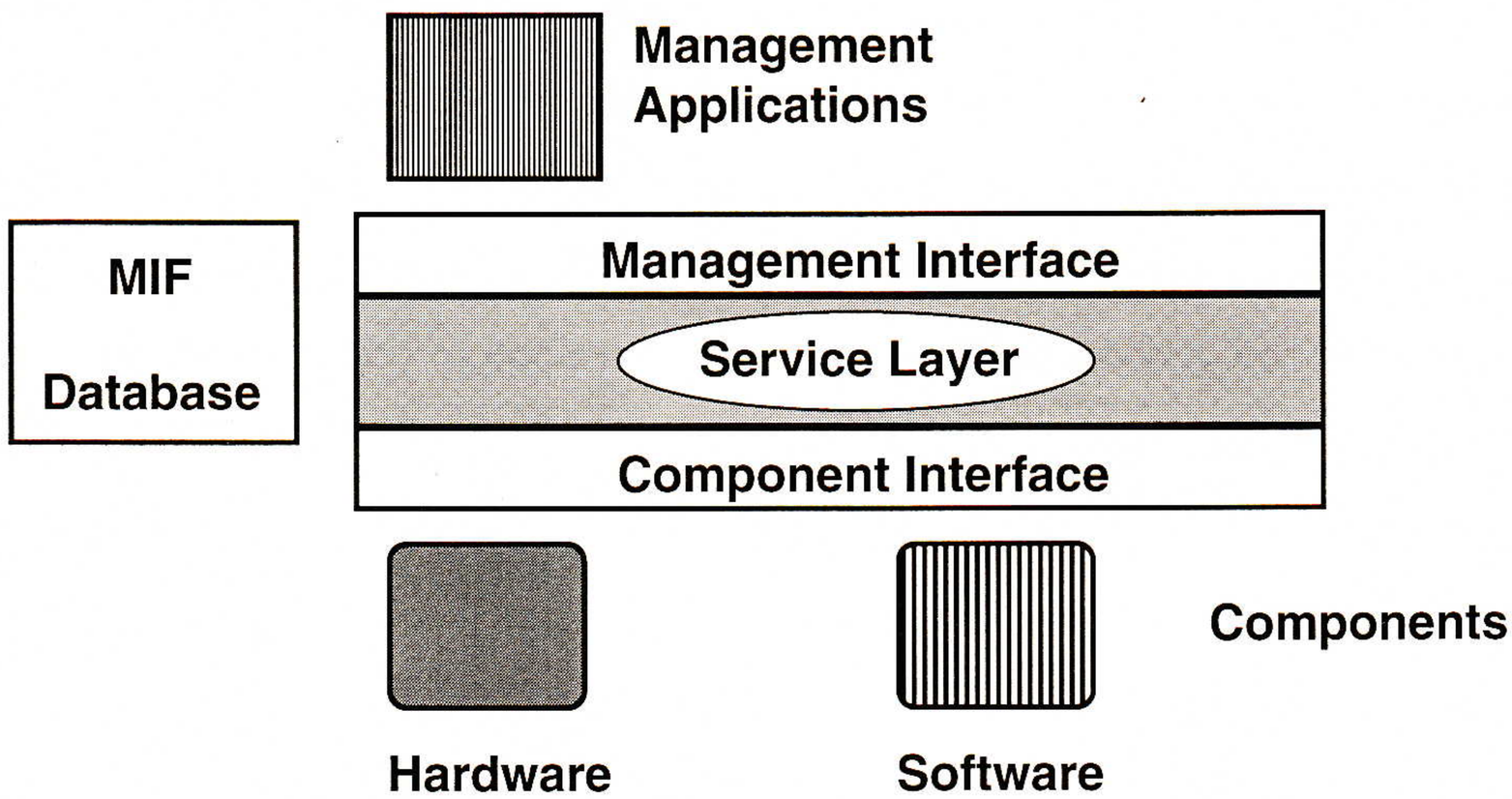


Figure 1: DMI Architecture

The Service Layer handles the run-time operation of DMTF functions, using operating systems-specific services as necessary to support desktop or server management. A fourth component, the MIF database, is an implementation-specific collection of component descriptions. The MIF database represents all components that are manageable at any given time.

Components

Components are manageable elements within a system, including processors, add-on boards, memory, storage, and applications. The *Component Instrumentation* (CI) is the software that carries out management functions with a component. The Component Instrumentation provides information about a component, changes the component's behavior as directed, or signals an asynchronous event when unexpected conditions occur. The CI provides access to component instrumentation for management applications.

continued on next page

DMTF *(continued)*

The MI provides a way for management applications to register for events, discover the types of components in the system, and access Component Instrumentation in order to carry out systems management tasks.

MIF

The *Management Information Format* (MIF) is a text-based description of each component in the system. Each component is defined with a unique name and identifier. A standard component identifier group has been defined by DMTF, which includes the manufacturer, the version, the serial number, a product identifier, time and date of the last installation, and a verification level, which is used to check that the component is working properly.

Each component definition contains a path identifier which provides file names where the actual Component Instrumentation code can be accessed through the Service Layer.

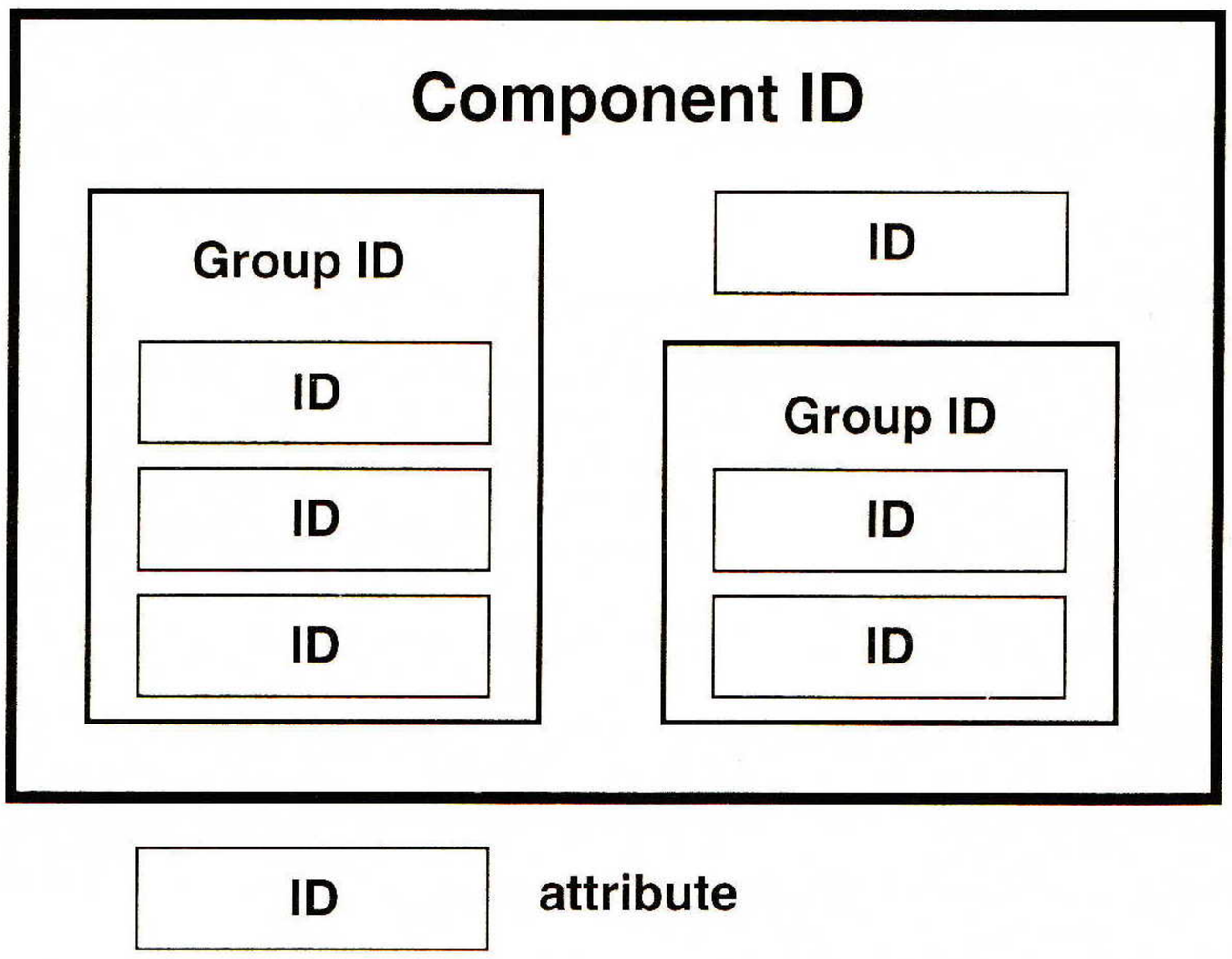


Figure 2: Component definition

Each component can be composed of groups, which are collections of attributes. Groups are an easy way of organizing attributes by categories such as those for configuration, performance, faults and other functional areas. Each attribute is represented by one of the data types, such as integer, counter, gauge, dates, strings, and octets. Attributes may also exist without a group membership.

Tables of attributes can also be created when there are multiple instances of a component in a system. For example, when there are multiple communications interfaces on a server.

The MIF definitions provide a unique identifier for each component, as well as unique identifiers for groups within components, and for attributes contained within each group. In this way, a simple hierarchical naming structure identifies any particular piece of information that the management application wishes to access or change. Tables are accessed by supplying a key or index instead of an attribute identifier.

MIF database

The MIF database contains all the component descriptions that have been registered with the Service Layer. Registration makes a component visible through the Management Interface to all management applications. Components can also be unregistered, which removes them from the MIF database and therefore from any further management activities through the Management Interface. The MIF database is a logical management information store that is implemented in a system-dependent way. The only requirement is that it preserve information across system failures or reboots.

Management Interface

The Management Interface provides a means for different management applications to receive events and access the Component Instrumentation as needed. The Management Interface moves blocks of data through the Service Layer to the appropriate Component Instrumentation. In this respect, the interface has not been based on procedural functions, but simply passes data asynchronously. Each management application is allowed to make multiple management requests, which are processed independently through the Service Layer and passed to the Component Interface.

The Service Layer

The Service Layer provides the run time management: controlling the flow of data blocks and events between Component Instrumentation and management applications. It handles buffering, error checking, integrity, and synchronization of all activities.

Data Blocks

All interactions for normal management operations are carried out through the exchange of structured data blocks. These blocks contain parameters and pointers to other memory areas, which are used to describe structures and to receive information from the Component Instrumentation. A standard set of commands has been defined by the DMTF and the command set will grow richer as new functions are incorporated into the DMTF definitions.

Management protocols

The DMTF has chosen to remain protocol-neutral, endorsing neither SNMP nor CMIP. Network management protocols can be used to manage DMTF-compliant systems across a network, with a *proxy agent* or translator at the Management Interface. The proxy agent translates network management protocol traffic into the appropriate DMTF commands and attribute identifiers. In a similar way, it translates information from the Component Instrumentation into the appropriate SNMP MIB variables before transferring them to a remote management application.

An example

Simpler, more intelligent installation is an example of the types of things that DMTF will provide. DMTF-compliant components will come equipped with a MIF description, management applications, and an installation program.

As a case in point, let's assume an administrator is installing a DMTF-compliant device in a desktop. After physically installing the appropriate add-on boards, the administrator runs the installation program. The installation program registers with the Management Interface and uses information from the MIF database to determine if other needed resources are available in the system. Versions are checked to insure that all elements will work together.

The MIF database makes installation programs much more intelligent; for example, the installer can find an unassigned interrupt or DMA channel and assigns them without conflicts with other components. The installer uses this information to update the appropriate system configuration files and component software.

continued on next page

DMTF *(continued)*

The component MIF is registered through the Service Layer and added to the set of manageable components in the MIF database. Users will be freed from the tedious details of installation and all the problems associated with resolving interrupt assignments, memory allocations, and so forth.

A component diagnostic program can be run as needed by the user or on a periodic basis. This program can run tests and make decisions about the ongoing viability of the component. Regular diagnosis spots trends, indicating potential problems before anything catastrophic actually occurs. The management application at that point could notify the user or a remote administrator that the component is showing signs of imminent failure. Other information, such as warranty information and vendor contacts is also supplied from the MIF database in order to facilitate the problem resolution or replacement process.

Current status

MIF definitions for printers, network adapters and PC systems will be available in July, September and late in 1994, respectively. The printer group has 11 members including HP, IBM, Abobe and Intel. The network adapter group has 6 members: 3Com, Intel, SMC, NSC, Asante and D-Link. The PC systems group also has 12 members including Compaq, Dell, Microsoft, Novell, IBM and AST.

New groups are forming throughout the next quarter. They will be building MIF definitions for servers, mass storage, modems and applications. Bringing applications under a common framework will improve the administrator's situation. Managing and monitoring applications will provide the means for delivering guaranteed service levels; managed applications will also make network management easier since abusive applications can be turned off.

Specifications, by themselves, do not offer any benefits. Early commitments from vendors are shaping up. Operating systems vendors that have agreed to a DMI interface in the future include IBM (OS/2), Microsoft (Chicago), Novell (NetWare and UNIXWare), Sun (Solaris) and Intel (DOS and Windows 3.X).

Intel and Microsoft have offered the LANDesk Manager and Hermes management applications that will be engineered for the DMI. SNMP agents are being provided by FTP, NetManage and SNMP Research. These agents will provide the DMI/MIB mapping, allowing remote management from an SNMP console.

Intel, SMC and 3Com will bring DMI-compliant network adapters late in the third quarter of 1994. Dell, AST and Compaq are expected to bring DMI-based products out by the end of the year.

Future prospects

There is already substantial interest in DMTF, a recent study indicated that over one-third of the system administrators surveyed are already planning to investigate and invest in DMTF. This is a very strong indication of interest, considering the products are not yet available. The long term prospects for DMTF look bright. Major vendors, such as Novell and Microsoft, have committed to implementing DMTF in future product releases. Many third-party management application vendors are attracted to a consistent interface across a wide number of systems supplied by Intel, IBM, Digital, Hewlett-Packard, and Sun.

There are several factors which influence the viability of DMTF in the marketplace. Administrators do not yet know the resource requirements that DMTF will impose on their systems. If DMTF requires large amounts of resident memory, as well as local storage and processing, it may prove to be unattractive from an economic or performance perspective.

Probably the strongest factor that will influence DMTF's eventual acceptance is the quality of the management applications and the manageable components that will be provided. Many vendors will have to cooperate in order to provide a wide array of DMTF products; a partial solution will not be acceptable. Component vendors must commit resources to designing Component Instrumentation and better management applications that fully exploit the possibilities of DMTF.

Third-party management application vendors also have a challenge: to take their top quality solutions and extend them across a wide spectrum of hardware and software environments. DMTF has attracted many vendors because good management is one of the key differentiators in an increasingly competitive marketplace. The next year will give potential buyers a good indication of whether the DMTF consortium can deliver on their promise.

References

[1] Desktop Management Task Force, "DMI Specification Version 1.0," Available from the DMTF, 2111 N.E. 25th Avenue, Hillsboro, OR 97124.

[2] *ConneXions,* Two *Special Issues* on Network Management and Network Security, Volume 3, No. 3, March 1989 and Volume 4, No. 8, August 1990.

[3] Case, J. D., Davin, J. R., Fedor, M. S., & Schoffstall, M. L., "Network Management and the Design of SNMP," *ConneXions,* Volume 3, No. 3, March, 1989.

[4] Case, J., McCloghrie, K., Rose, M. T., Waldbusser, S., "The Simple Management Protocol and Framework: Managing the Evolution of SNMP," *ConneXions,* Volume 6, No. 10, October 1992.

[5] Rose, M. T., "Network Management is Simple: You just need the 'Right' Framework." In *Integrated Network Management, II,* Iyengar Krishan and Wolfgang Zimmer, editors, pages 9–25, North Holland, April 1991.

[6] Rose, M.T., *The Simple Book: An Introduction to Internet Management,* Second Edition, Prentice-Hall, ISBN 0-13-177254-6, 1993.

[7] Case, J. D., McCloghrie, K., Rose, M. T. & Waldbusser, S. L., "Introduction to version 2 of the Internet-standard Network Management Framework," RFC 1441, April 1993.

On-line discussions of DMTF specifications and implementation issues are carried out on the mailing list:

`dmtf-info@Sun.com`.

Subscribe in the usual way.

JOHN McCONNELL is President of McConnell Consulting—a company that focuses on systems and network management issues in the internetworking area. John has led consulting and training assignments throughout the world for vendors and end-users. He is completing a book on systems and network managementt for Prentice-Hall this year, and he teaches a tutorial for NetWorld+Interop focusing on practical management solutions. E-mail: `johnmc@mcconnell.com`

Announcement and Preliminary Call for Papers

The 5th *USENIX UNIX Security Symposium* will be held June 5–7, 1995 at the Salt Lake City Marriott Hotel in Salt Lake City, Utah. The event is sponsored by the USENIX Association, the UNIX and Advanced Computing Systems Professional and Technical Association, in cooperation with The Computer Emergency Response Team (CERT), IFIP WG 11.4 (pending), and UniForum (pending).

Goal

The goal of this symposium is to bring together security practitioners, researchers, system administrators, systems programmers, and others with an interest in computer security as it relates to networks and the UNIX operating system.

Format

This will be a 3 day, single-track symposium. It will consist of tutorials, refereed and invited technical presentations, and panel sessions. The first day will be devoted to tutorial presentations. Two days of technical sessions will follow the tutorials. The one-day tutorial program (June 5th) is designed to address the needs of both technical and management attendees. The tutorials will supply overviews of various security mechanisms and policies. Each will provide specifics to the system and site administrator for implementing numerous local and network security precautions, firewalls, and monitoring systems.

Keynote

The keynote address by Stephen T. Walker, Founder and President of Trusted Information Systems, will begin the technical sessions program. Mr. Walker will speak on information security and privacy in computing. Mr. Walker is an electronics engineer and computer systems analyst with over 25 years of experience in system design and program management; particularly extensive is his experience with the design and implementation of large scale computer networks and information systems. He is nationally recognized for his pioneering work on the DoD Computer Security Initiative, the establishment of the National Computer Security Center, and the formation of the Defense Data Network. He is a member of the Computer System Security and Privacy Advisory Board, established by the Computer Security Act of 1987.

Technical sessions

The technical sessions program, in addition to presentations of refereed papers, will include invited talks, and possibly panel sessions. There will also be two evenings available for Birds-of-a-Feather sessions (BoFs) and Works-in-Progress Reports (WiPs). The program committee invites you to submit proposals, ideas, or suggestions for these presentations; your suggestions may be submitted to the program chair via e-mail to: `security@usenix.org` or by post to the address given below.

Papers that have been formally reviewed and accepted will be presented during the symposium and published in the symposium proceedings. Proceedings of the symposium will be published by USENIX and will be provided free to technical session attendees; additional copies will be available for purchase from USENIX.

Topics

Presentations are being solicited in areas including but not limited to:

- User/system authentication
- File system security
- Network security
- Security and system management
- Security-enhanced versions of the UNIX operating system
- Security tools
- Security incident investigation and response
- Computer misuse and anomaly detection

- Security in heterogeneous environments
- Configuration management to support security
- Security-related testing methods
- Case studies

Submissions

Full papers should be 10 to 15 pages. Instead of a full paper, authors may submit an extended abstract which discusses key ideas. Extended abstracts should be 5–7 pages long (about 2500–3500 words), not counting references and figures. The body of the extended abstract should be in complete paragraphs. The object of an extended abstract is to convince the reviewers that a good paper and presentation will result. All submissions will be judged on originality, relevance, and correctness. Each accepted submission will be assigned a member of the program committee to act as its shepherd through the preparation of the final paper. The assigned member will act as a conduit for feedback from the committee to the authors.

Please accompany each submission by a cover letter stating the paper title and authors along with the name of the person who will act as the contact to the program committee. Please include a surface mail address, daytime and evening phone number, and, if available, an e-mail address and fax number for the contact person. If you would like to receive detailed guidelines for submission and examples of extended abstracts, you may send e-mail to: `securityauthors@usenix.org` or telephone the USENIX Association office at +1 510 528 8649. The UNIX Security Symposium, like most conferences and journals, requires that papers not be submitted simultaneously to another conference or publication and that submitted papers not be previously or subsequently published elsewhere. Papers accompanied by "non-disclosure agreement" forms are not acceptable and will be returned to the author(s) unread. All submissions are held in the highest confidentiality prior to publication in the Proceedings, both as a matter of policy and in accord with the U.S. Copyright Act of 1976.

Please send one copy of a full paper or an extended abstract to the program committee via *two* of the following methods. All submissions will be acknowledged.

- E-mail (*PostScript* or ASCII) to: **`security@usenix.org`** ; preferred
- Alternate Method: postal delivery to
  Fred Avolio
  Trusted Information Systems
  3060 Washington Road
  Glenwood, MD 21738
- Fax: +1 301 854 5363

Registration information

Materials containing all details of the technical and tutorial programs, registration fees and forms, and hotel information will be available beginning in March 1995. If you wish to receive the registration materials, please contact USENIX at:

USENIX Conference Office
22672 Lambert Street, Suite 613
Lake Forest, CA 92630
Phone: +1 714 588 8649 • Fax: +1 714 588 9706
E-mail: `conference@usenix.org`

Important dates

Extended abstracts due:	February 13, 1995
Program Committee decisions made:	March 8, 1995
Camera-ready final papers due:	May 1, 1995

Announcement and Preliminary Call for Papers

Introduction

The Second USENIX *Symposium on Mobile and Location-Independent Computing* will be held April 10–11, 1995 in Ann Arbor, Michigan. The symposium will provide a major opportunity for researchers and practitioners in this rapidly growing field to exchange ideas and present results of their work.

The First Mobile Computing Symposium, held in Boston in August 1993, generated a great deal of interest from the UNIX and mobile computing communities. Since that time, mobile computing has become an even hotter topic, with the size, cost, and power requirements of the equipment going down. The FCC has announced a plan to auction radio spectrum for use of mobile devices, and the Internet Engineering Task Force (IETF) is in the process of standardizing protocols for mobile TCP/IP, including roaming capabilities. Mobile computers are the fastest growing segment of the PC market, airlines are scrambling to provide network connectivity on board, and terminal rooms at computer conferences routinely provide network taps for users who bring their own computers.

Format

The 1995 symposium is a single-track symposium offering two days of refereed paper presentations. The symposium will also include panels, Work-in-Progress reports, Birds-of-a-Feather sessions, and a Keynote speaker. Formally reviewed papers, presented during the symposium, will be published in the symposium proceedings. Proceedings will be distributed free to attendees and later will be available for purchase from the USENIX Association.

Topics

We seek original and innovative papers about current developments in mobile and location-independent computing. We are especially interested in reports on practical experiences with mobile systems. The Mobile Computing Symposium will address a wide range of issues and ongoing developments, including, but not limited to:

- Applications for the mobile user
- Navigation and positioning (GPS, etc.)
- Security, especially in wireless environments or when away from home
- Caching/disconnected operation of applications and file systems
- Communications Protocols, including mobile TCP/IP
- Wireless communications (CDPD, CDMA, GSM, Ardis/RAM, cellular modem, etc.), and how they relate to and interact with operating systems and applications
- Portable and mobile computing equipment

Submission guidelines

Submission of an extended abstract of 1,500–2,500 words (9,000–15,000 bytes or 3–5 pages) is recommended. Shorter abstracts run a significant risk of rejection as there will be little on which the program committee can base an opinion. Extended abstracts should be sent to Jim Rees at the address below. If you would like to receive detailed guidelines for submission and examples of extended abstracts, you may contact the USENIX Association office by telephone at +1 510 528-8649 or e-mail to `mobile2authors@usenix.org`

For administrative reasons (not blind reviewing), each submission should include a separate page or e-mail message giving the title of the paper, the names and affiliations of the authors, and the name of the author who will act as the contact person for the program committee. For the contact person, also include a daytime telephone number, postal address, e-mail address and FAX number if possible.

USENIX symposia, like most symposia and journals, require that papers not be submitted simultaneously to more than one conference or publication and that submitted papers not be previously or subsequently published elsewhere. Papers accompanied by "non-disclosure agreement" forms are not acceptable and will be returned to the author(s) unread. All submissions are held in the highest confidentiality prior to publication in the Proceedings, both as a matter of policy and in accord with the U.S. Copyright Act of 1976.

Important dates

Extended abstracts due: January 2, 1995

Notification to authors: January 23, 1995

Camera-ready final papers due: March 6, 1995

More program information

For questions about refereed paper submissions and other program concerns, contact the Program Chair:

Jim Rees
CITI
University of Michigan
519 West William
Ann Arbor, Michigan 48103
Internet: `Jim.Rees@umich.edu`
Phone: +1 313 763-4174
Fax: +1 313 763-4434

Registration information

Materials containing all details of the technical and tutorial programs, registration fees and forms, and hotel information will be available beginning in February 1995. If you wish to receive the registration materials, please contact USENIX at:

USENIX Conference Office
22672 Lambert Street, Suite 613
Lake Forest, CA 92630
Phone: +1 714 588-8649
Fax: +1 714 588-9706
Internet: `conference@usenix.org`

For questions about your subscription please call our customer service hotline: 1-800-575-5717 or +1 502-493-3217 outside the USA. This is the number for our subscription agency, the Cobb Group. Their fax number is +1 502-491-8050. The mailing address for subscription payments is P.O. Box 35840, Louisville, KY 40232–9496.

Write to *ConneXions*!

We'd love to hear your comments, suggestions and questions about anything you read in *ConneXions*. Our editorial address is given below. Use this address for letters to the Editor, questions about back issues etc.:

ConneXions—The Interoperability Report
303 Vintage Park Drive
Suite 201
Foster City, CA 94404–1138
USA
Phone: +1 415-578-6900 or 1-800-INTEROP (Toll-free in the USA)
Fax: +1 415-525-0194
E-mail: `connexions@interop.com`

CONNEXIONS
303 Vintage Park Drive
Suite 201
Foster City, CA 94404-1138
Phone: 415-578-6900
FAX: 415-525-0194

ADDRESS CORRECTION
REQUESTED

FIRST CLASS MAIL
U.S. POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 1